The I-7530 RS-232/CAN Converter

User's Manual

Warranty

All products manufactured by ICP DAS are under warranty regarding defective materials for a period of one year from the date of delivery to the original purchaser.

Warning

ICP DAS assumes no liability for damages resulting from the use of this product. ICP DAS reserves the right to change this manual at any time without notice. The information furnished by ICP DAS is believed to be accurate and reliable. However, no responsibility is assumed by ICP DAS for its use, or for any infringements of patents or other rights of third parties resulting from its use.

Copyright

Copyright 1997 by ICP DAS. All rights are reserved.

Trademark

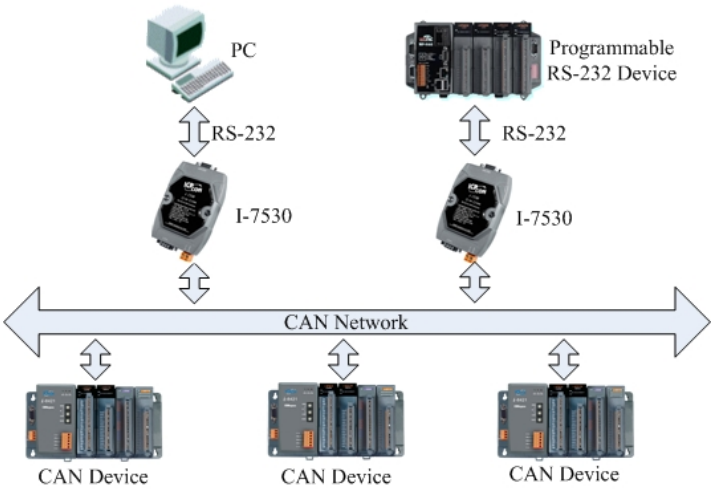
The names used for identification only may be registered trademarks of their respective companies.

Table of Contents

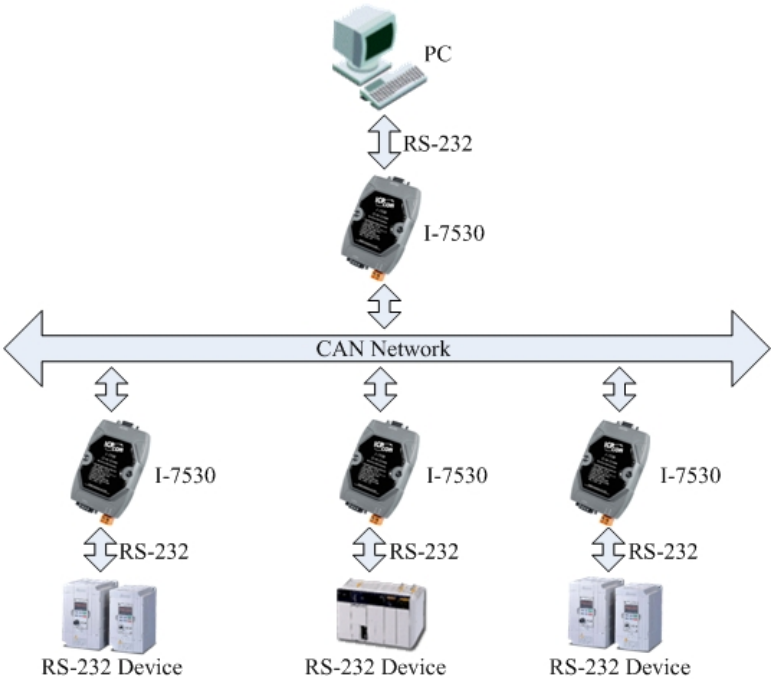
1. Introduction 3
1.1 Features 4
1.2 Specifications 4
2. Hardware 6
2.1 Block Diagram 6
2.2 Pin Assignment 6
2.3 Hardware connection 8
2.4 Terminator resistor settings 9
2.5 Init/Normal Dip-switch 10
2.6 LED Indication 11
2.7 Cable selection 12
3. Software Utility 13
3.1 How to configure the module parameters 14
3.2 How to set the Acceptance Code and Mask 18
3.3 Test I-7530 on CAN network (Only for normal mode) 19
3.4 Pair Connection Mode Description 21
4. Command list 26
4.1 tIIILDD…[CHK]<CR> 28
4.2 TIIIL[CHK]<CR> 28
4.3 eIIIIIIIILDD…[CHK]<CR> 29
4.4 EIIIIIIIIL[CHK]<CR> 29
4.5 S[CHK]<CR> 30
4.6 C[CHK]<CR> 32
4.7 P0BBDSPAE[CHK]<CR> 32
4.8 P1B [CHK]<CR> 34
4.9 RA[CHK]<CR> 35
4.10 General Error code for all command 35
5. Troubleshooting 36

1. Introduction

CAN (Controller Area Network) is a serial bus control protocol especially suited to structure intelligent industry devices networks and build smart automatic control systems. By using I-7530, some programmable RS-232 devices such as PC, I-8000, WinPAC-8000 embedded controllers, can be the master of a CAN network. Therefore, they can control or monitor the CAN devices via the I-7530 module.



Moreover, we expand the functionalities of I-7530 for some special application. In pair connection mode, I-7530 can be used to connect PC with other RS-232 devices at the same time. The application architecture may be as follows.



1.1 Features

- Microprocessor inside with 20MHz
- CAN interface connector: D-Sub 9-pin
- Phillip 82C250 CAN transceiver
- Provide One CAN and RS-232 Port
- Built in CAN/RS-232 Converter firmware
- Max transmission speed up to 1M bps for CAN and 115.2K bps for RS-232 interface
- Provide Software configuration for CAN and RS-232 communication
- Max transmission distance over 1000m based on CAN specification
- Support both CAN 2.0A and CAN 2.0B
- Build-in jumper to provide 120 ohm terminal resistor or not in CAN network
- Provide power, data flow and error LED indicator for CAN and RS-232
- Built-in a Dual-Watchdog function in the module
- Support transparent communication mode
- Provide the transparent communication between the RS-232 devices via CAN bus.
- Enable different RS-232 devices into an individual group in CAN bus network (Full-duplex communication mode of RS-232 devices is not supported)

1.2 Specifications

RS-232 specifications:

- RS-232 interface connector: D-sub 9-pin female
- RS-232 Baud Rate: 110, 150, 300, 600, 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200

CAN specifications:

- CAN interface connector: D-sub 9-pin male
- CAN Baud Rate: 10K, 20K, 50K, 100K, 125K, 250K, 500K, 800K and 1M bps
- Isolation voltage: 3000Vrms on the CAN side

Power requirement:

- Unregulated +10V DC ~ +30V DC

- Power reverse protection, Over-Voltage brown-out protection
- Power consumption: 1W

Module specifications:
- Dimensions: 123mm x 72mm x 33mm
- Operating temperature: -25 to 75°C (-13 to 167°F)
- Storage temperature: -40 to 80°C (-40 to 176°F)
- Humidity: 5 to 95%, non-condensing
- LEDs: ON LED for power and data flow
  ERR LED for error

Software Utility tool:
- CAN bus Baud Rate configuration
- CAN acceptance filter configuration
- Setting for CAN 2.0A or 2.0B working mode
- Setting for I-7530 communication mode (normal mode or pair connection mode)
- Setting the Baud Rate and data format on RS-232
- Setting the RS-232 communication with checksum function or not
- Provide a response message or not when the ERR LED is turned on
- Provide a quick testing function for transmitting/receiving CAN messages

Application:
- Factory Automation
- Building Automation
- Home Automation
- Control system
- Monitor system
- Vehicle Automation

2. Hardware

2.1 Block Diagram

Figure 1 is a block diagram illustrating the functions on the I-7530 module. It provides the 3000Vrms Isolation in the CAN interface site. And hardware media in RS-232 interface is only adopted 3-wire connection.

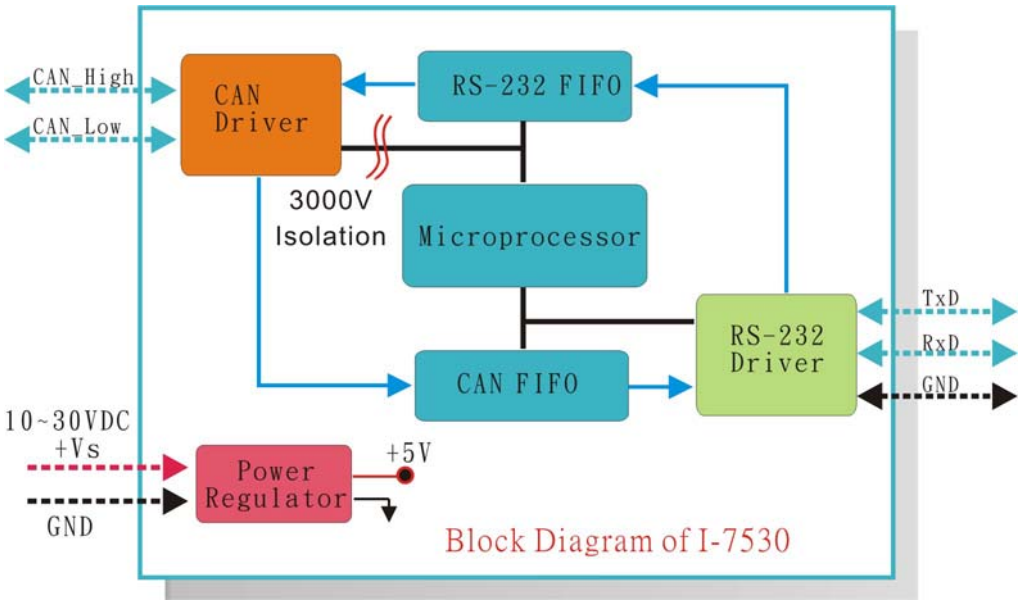


Figure 1: Block diagram of I-7530

2.2 Pin Assignment

Table 1: RS-232 DB9 Female Connector (CN1)

Terminal	3-wire RS-232
1	Not Connect
2	TXD
3	RXD
4	Not Connect
5	GND
6	Not Connect
7	
8	
9	

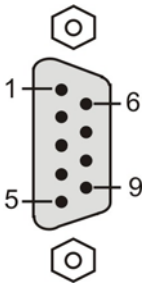


Table 2: CAN DB9 Male Connector (CN2)

Terminal	2-wire CAN
1	Not Connect
2	CAN Low
3	Not Connect
4	
5	
6	
7	CAN High
8	Not Connect
9	

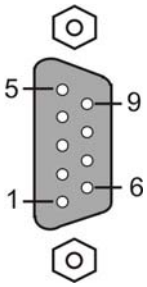


Figure 2: Pin Assignment of the I-7530

2.3 Hardware connection

The RS-232 port on the I-7530 (DB9 female) is inserted directly into a PC's COM serial port or via a cable to the Host system.

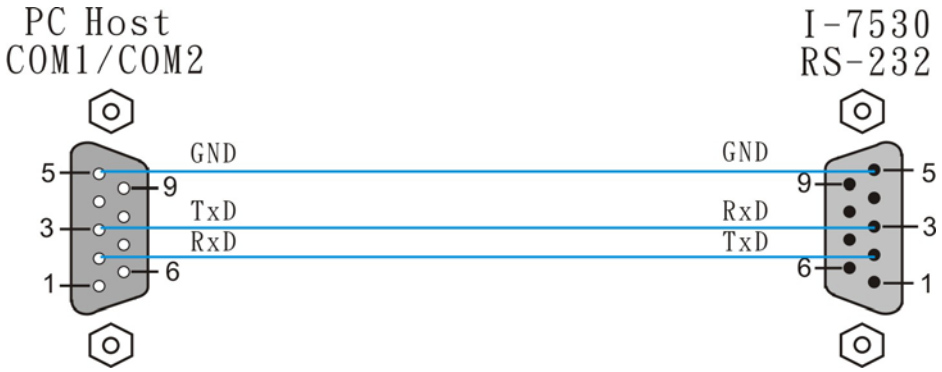


Figure 3: RS-232 connection

The pin assignment of the CAN port on the I-7530 (DB9 male) defined in both the CANopen DS102 profile and in appendix C of the DeviceNet specifications. It is the standard pin assignment for CAN.

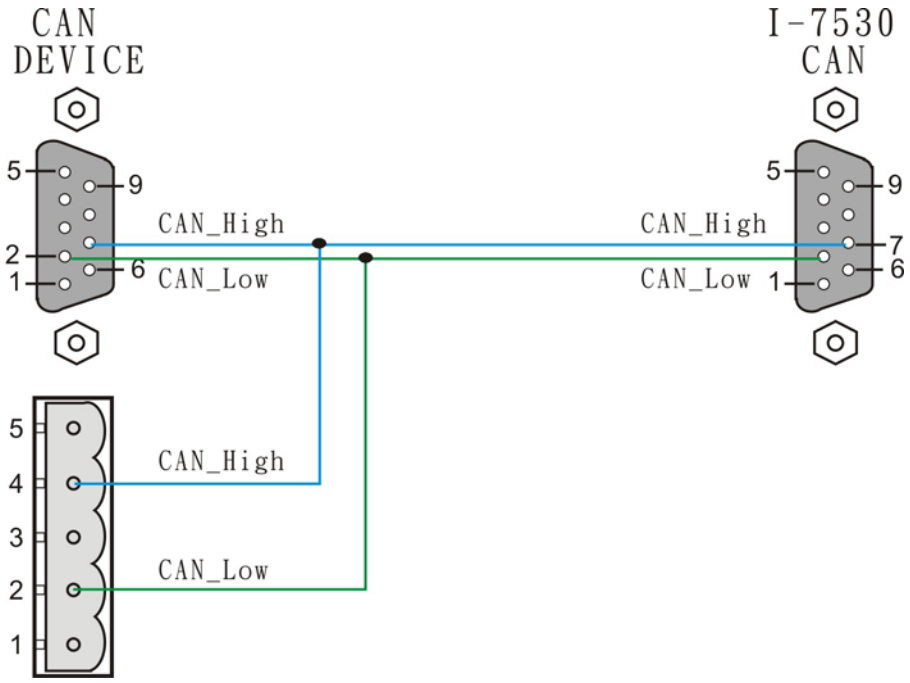


Figure 4: CAN connection

2.4 Terminator resistor settings

According to the ISO 11898 specifications, the CAN Bus network must be terminated by two termination resistances (120Ω) for proper operation, as shown in the below figure.

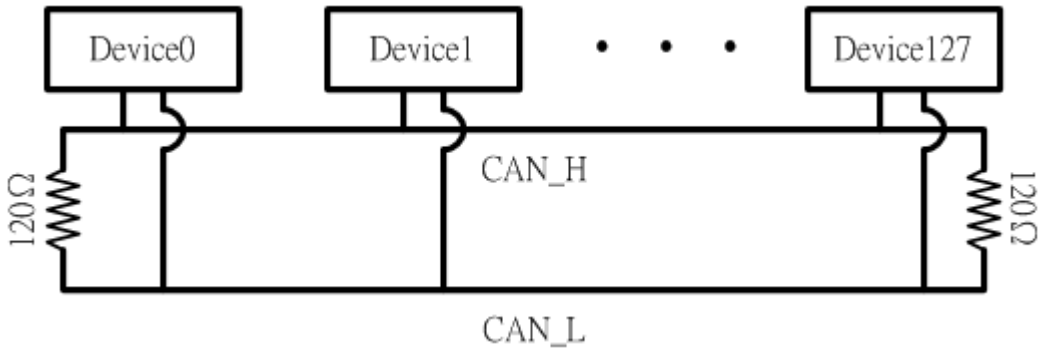


Figure 5: Terminator resistor

Therefore, the I-7530 module supplies a jumper for users to connect the terminator resistor or not. If users want to use this terminator resistor, please open the I-7530 cover and use the JP3 jumper to activate the 120 Ω terminator resistor built in the system, as in the following figure. Note that the default setting is active.

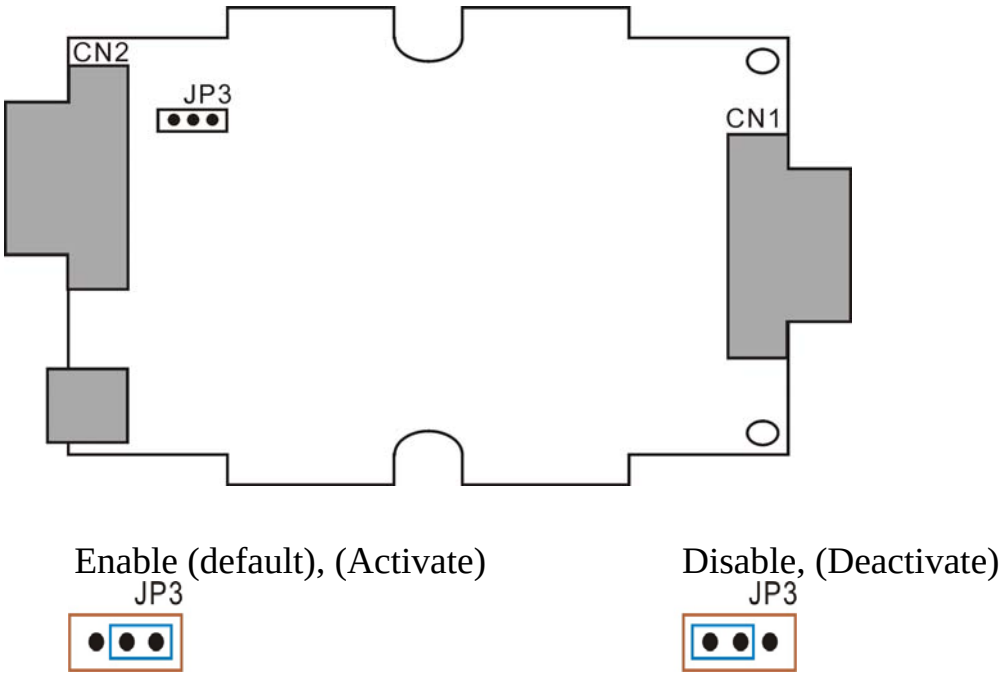


Figure 6: JP3 positions

2.5 Init/Normal Dip-switch

On the back of the I-7530 module, there is a dip-switch used for setting the operation or configuration mode of the module function. In the normal situation, the user needs to first make a configuration in order to provide the correct function when the module works in the operation mode. The following steps show how to use this dip-switch.

(1) Configuration mode: Please first set the dip-switch to the “Init” (Initial) position. Then the I-7530 will work in the configuration mode after the power for the module has been turned on. In this case, users can configure the communication parameters of the I-7530 module by using the I-7530 Utility tool.
(2) Operation mode: After the configuration, users need to set the dip-switch to the “Normal” position. Users need to turn the power off then on again so that they can use the I-7530 in the operation mode. Once you have completed your configurations and have switched to the operation mode, then messages can pass between the CAN and the RS-232. This transmission process depends on the configuration parameters that the users have configured. That is, the I-7530 functions as a RS-232/CAN converter.

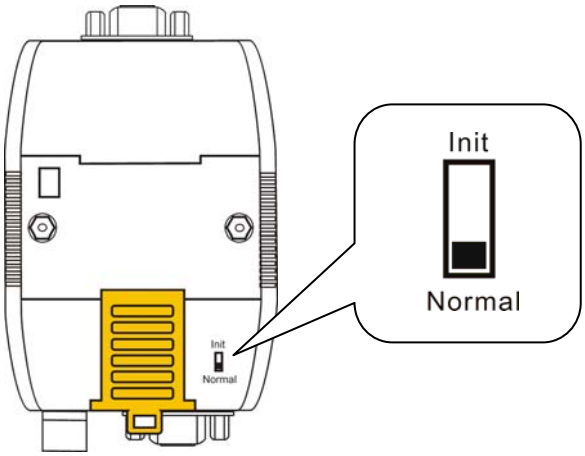


Figure 7: Dip-switch of I-7530

2.6 LED Indication

The I-7530 provides the Converter function between the RS-232 port and the CAN port. It can handle both an 11-bit and 29-bit ID format according to whether it is a CAN 2.0A or 2.0B. In order to provide high performance when converting data, the I-7530 has built in software FIFO queues, which include 1000 CAN data frames. It also provides two LEDs to indicate to users what situation the I-7530 is in. The ON LED indicator establishes the power and operation situation simultaneously. If the module is working in the operation mode, the ON LED is always turned on. However, when the module is working in the configuration mode, the ON LED will flash approximately once per second. The ERR LED is used for demonstrating an error that has occurred. That is, the ERR LED is normally turned off when the module works in a good condition. On the other hand, when the BUS OFF error indicator of the CAN controller has been alerted, the ERR LED will turn on. In this case, users can only reboot the I-7530 module in order to turn off the ERR LED. If the CAN/RS232 FIFO overflows, the ERR LED will also be turned on. Furthermore, If users want to check what the error situation is they must send the command string “S[CHK]<CR>”. Depending on the result, users will need to either reboot the I-7530 or use the command string “C[CHK]<CR>” to clear the FIFO flag, which will then turn off the ERR LED.

Figure 8: LED position

Table 3: LED indication

LED Name	I-7530 Condition	LED Status
ON LED	Configuration Mode	Flashes once per second
	Operation Mode	Turn on when no data needs to be send to CAN bus
		Flash once when sending a CAN message
ERR LED	Some errors occurred	Always turned on

2.7 Cable selection

The CAN bus is a Balanced (differential) 2-wire interface running over either a Shielded Twisted Pair (STP), Un-shielded Twisted Pair (UTP), or Ribbon cable. The CAN-L and CAN-H Wire start on one end of the total CAN network that a terminator of 120 Ohm is connected between CAN-L and CAN-H. The cable is connected from CAN node to CAN node, normally without or with short T connections. On the other end of the cable again a 120Ω(Ohm) terminator resistor is connected between the CAN lines. How to decide a cable type, cable length, and terminator depends on the baud rate in the CAN BUS network, please refer to the following table 4.

Figure 9: Un-shielded Twisted Pair (UTP)

Table 4: Cable selection

Bus speed	Cable type	Cable resistance/m	Terminator	Bus Length
50k bit/s at 1000m	0.75~0.8mm2 18AWG	70 mOhm	150~300 Ohm	600~1000m
100k bit/s at 500m	0.5~0.6 mm2 20AWG	< 60 mOhm	150~300 Ohm	300~600m
500k bit/s at 100m	0.34~0.6mm2 22AWG, 20AWG	< 40 mOhm	127 Ohm	40~300m
1000k bit/s at 40m	0.25~0.34mm2 23AWG, 22AWG	< 40 mOhm	124 Ohm	0~40m

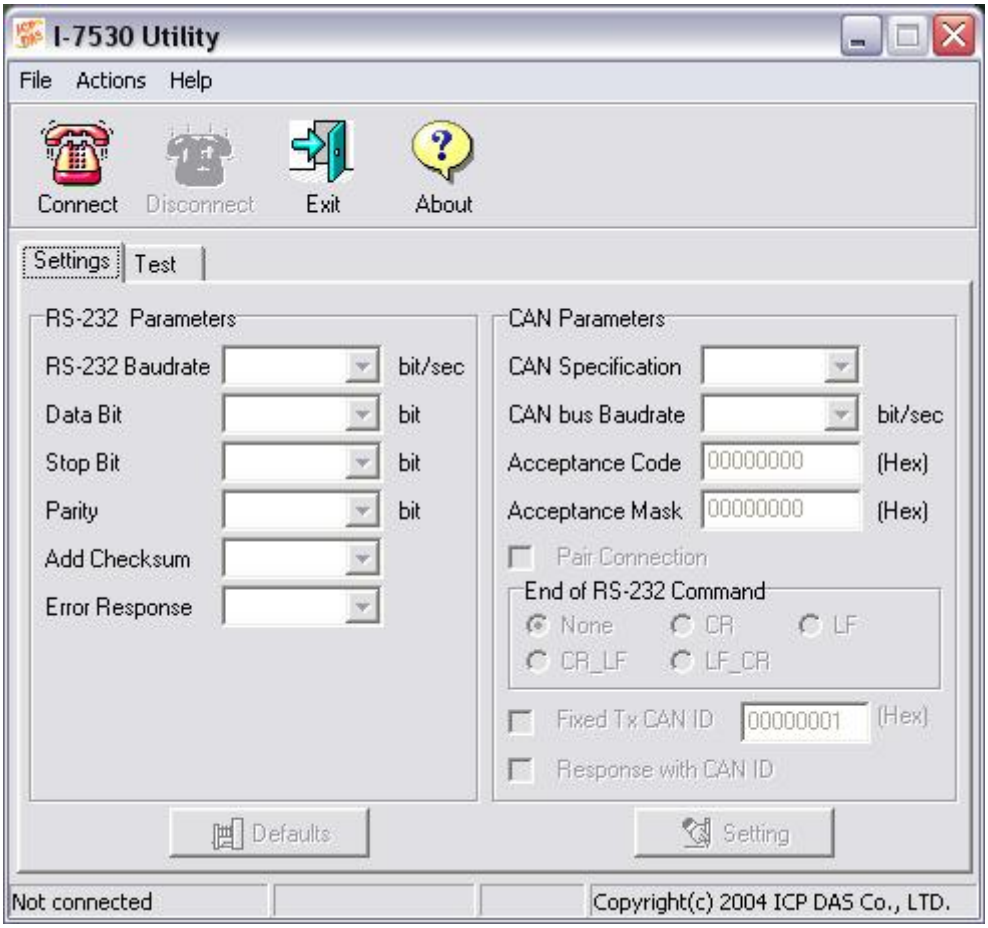
Note: The AWG means a standard method used to measure wire. The numbering system works backwards from what people would think, the thicker (heavier) the wire, the lower the number. For example: a 24AWG wire is thicker/heavier than a 26AWG wire.

3. Software Utility

This section will show you how to configure the I-7530 and test it by using I-7530 Utility. Users can download I-7530 Utility software from the ICP DAS web site:

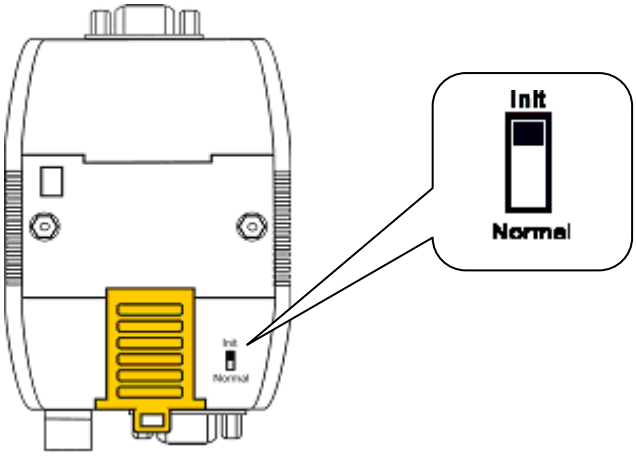
http://www.icpdas.com/products/Remote_IO/can_bus/i-7530.htm

or get this information from the ICP DAS field_bus CD-ROM. The I-7530 Utility tool can be used to configure the operation status of the CAN and RS-232 communications. Also it can be used to transmit or receive a CAN message for simple testing. To start the “I-7530 Utility”, please run the I7530.exe file. The screenshot of the startup screen for this Utility is given in the below figure.

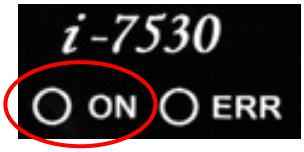


3.1 How to configure the module parameters

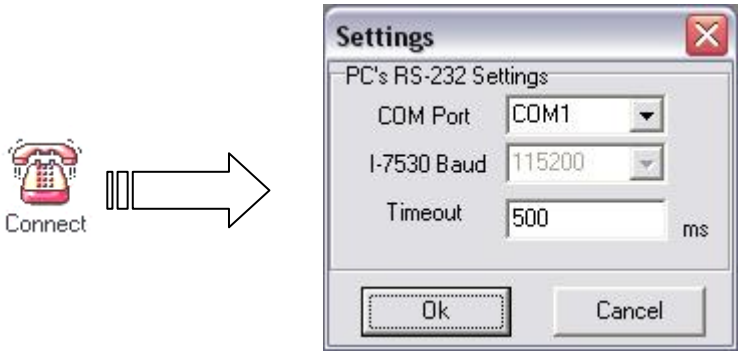
1. Set the Init/Normal switch to the Init position, which is located at the back of the I-7530 module.



2. Connect the power source (the 10~30 DC volts) into the I-7530 module.
3. The I-7530 module's ON LED will flash approximately once per second. That means the I-7530 module is working in the configuration mode.

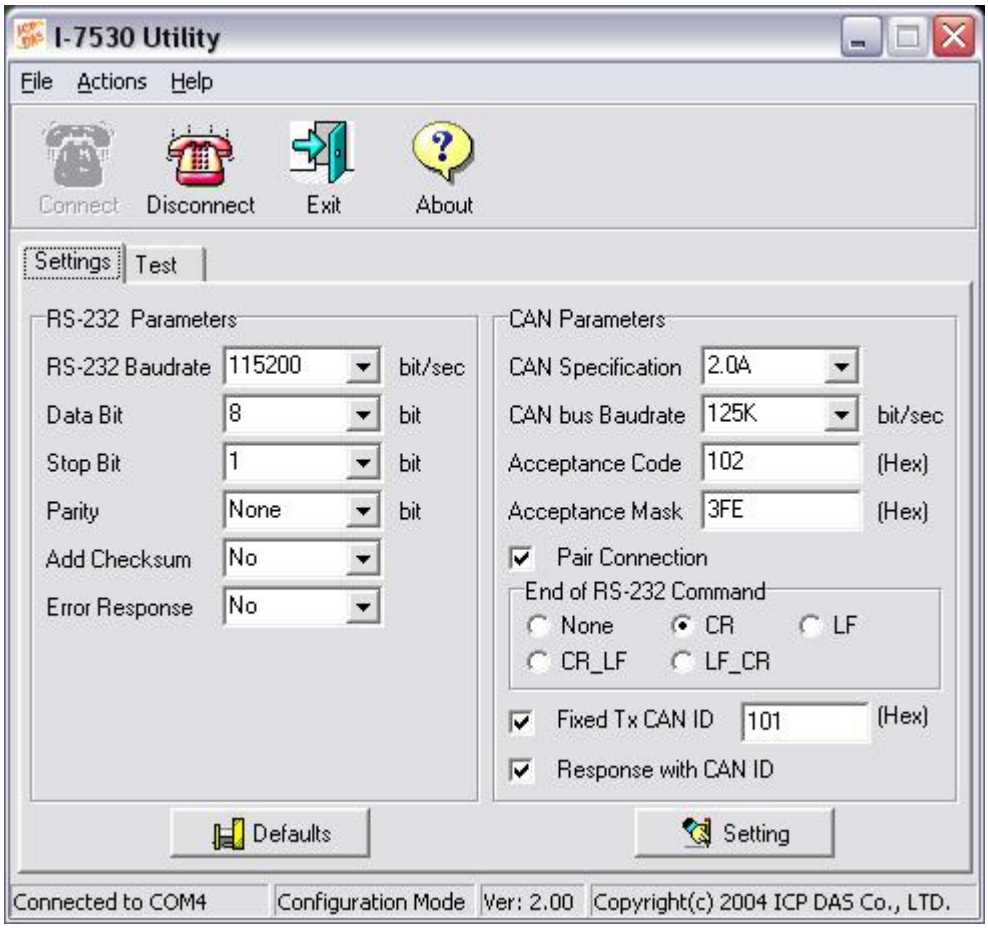


4. Run the I-7530's Utility software after connect the PC COM port and the I-7530 COM port by a RS-232 cable.
5. Click the "Connect" icon on the I-7530 Utility tool bar. The setting frame will be popped up. Select which PC COM port will be used to connect to the I-7530 COM port, as shown in the following figure. Then click the "OK" button.

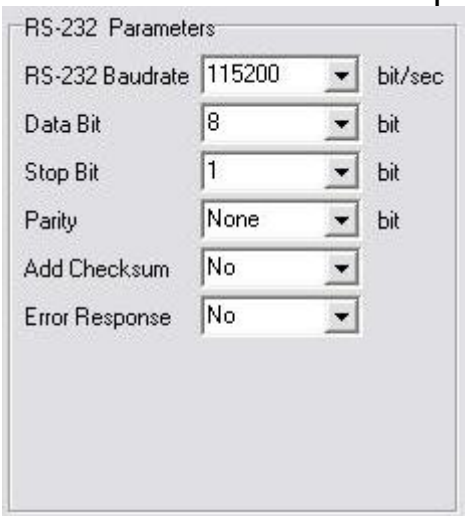


Note: If the I-7530 is in the Init mode or users choose the "Settings" tab, they can only communicate with each other by using 115200 Baud.

6. Then the I-7530 configuration window will be brought out. The I-7530 Utility will show the communication information from the I-7530 module in the window.



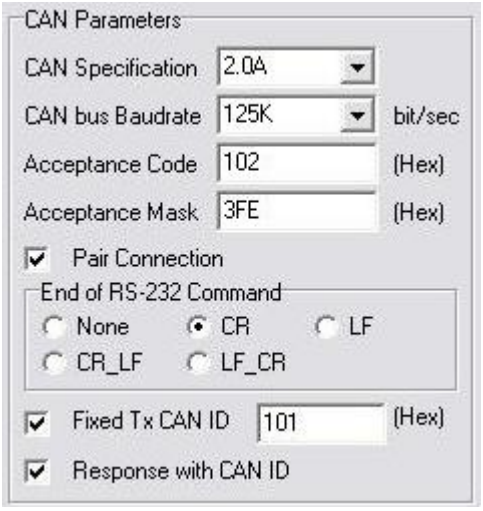
7. Set the RS-232 parameters for I-7530 RS-232 port.



Note: 1. When the function “Add Checksum” is set to “Yes”, users need to communicate to I-7530 with checksum mechanism.
2. If the “Error Response” is set to “Yes”, the error code will be

responded when the incorrect communication commands are sent to I-7530.

8. Set the CAN parameters for I-7530 CAN port.



Note:
1. About how to set the Acceptance Code and Acceptance Mask, please refer to the section 3.2.
2. If the “Pair Connection” is checked, the functions of “End of RS-232 command”, “Fixed Tx CAN ID” , and “Response with CAN ID” are useful.
3. When “Pair Connection” is checked, I-7530 is into pair connection mode. If it is unchecked, I-7530 is normal mode. In pair connection mode, all commands written to I-7530 COM port will transfer to the CAN bus directly. For more detail information about pair connection mode, please refer to the section 3.4.

9. Once users have finished all parameter settings, please click the “Setting” button to store the communication parameters into the EEPROM on the I-7530.

Note: If users click the “Defaults” icon, all of the CAN and RS-232 communication parameters on the I-7530 will be set to the default values, which are:

RS232:	RS-232 Baudrate	= 115200
	Data Bit	= 8
	Stop Bit	= 1
	Parity	= None
	Add Checksum	= No
	Error Response	= No
CAN:	CAN Specification	= 2.0A
	CAN bus Baud rate	= 125K
	Acceptance Code	= 000
	Acceptance Mask	= 000
	Pair Connection	= None

3.2 How to set the Acceptance Code and Mask

The acceptance code and acceptance mask are used for filtering unnecessary CAN messages through RS-232 port. If users want to prevent the unnecessary CAN message through RS-conversions, they need to set the parameters of acceptance code and mask.

Acceptance Code: The CAN ID bits that you want to get.
Acceptance Mask: The CAN ID bits that you want to filter.

In the acceptance code, the bit value '1' means that you want to get this CAN ID bit. And in the acceptance mask, the bit value '1' means that you want to filter this CAN ID bit. For Examples:

(1) If users want to get all the messages on the CAN bus, the setting must be as follow:

In CAN 2.0A: Acceptance Code: Don't care
Acceptance Mask: 000

(2) If users want to get the message of CAN ID "123" (Hex) and filter all the other CAN message, the setting must be as follow:

In CAN 2.0A: Acceptance Code: 123 (001 0010 0011 bit)
Acceptance Mask: 7FF (111 1111 1111 bit)

(3) If users want to get the message of CAN ID from "100" to "12F" (Hex) and filter all the other CAN message, the setting must be as follow:

In CAN 2.0A: Acceptance Code: 10X (001 00XX XXXX bit) X: don't care
Acceptance Mask: 7C0 (111 1100 0000 bit)

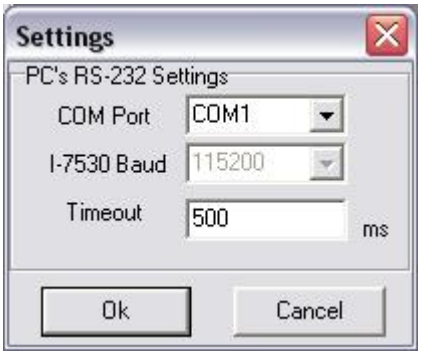
(4) The settings of CAN 2.0A and CAN 2.0B are the same. The different between CAN 2.0A and CAN 2.0B is the CAN ID bits.

In CAN 2.0A: number of CAN ID bits is 11.
In CAN 2.0B: number of CAN ID bits is 29.

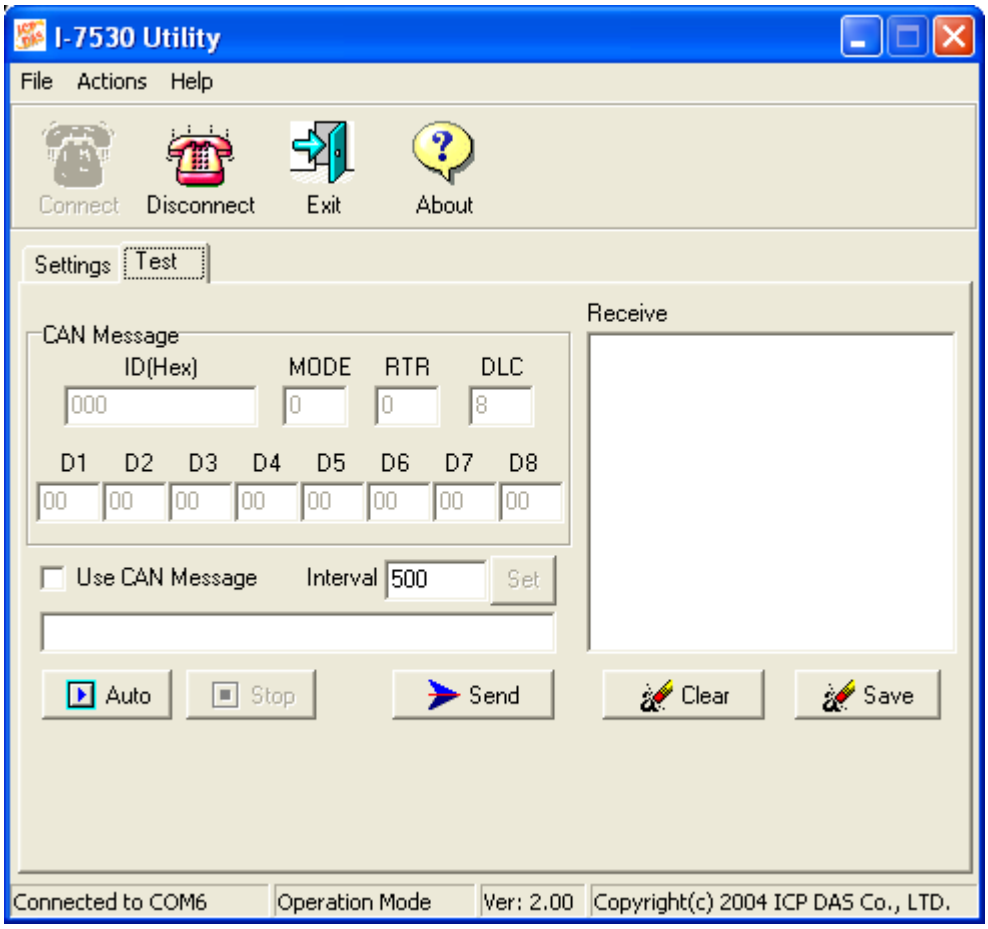
3.3 Test I-7530 on CAN network (Only for normal mode)

The following procedure will guide you to learning how to transmit/receive CAN messages to/from other devices/PCs by using the I-7530 converter. Note that users can only employ this test function for the system if they are using the default communication parameters for the RS-232 as provided above.

1. Set the Init/Normal switch to the Normal position, which is found at the back of the I-7530 module.
2. Connect the I-7530's CAN port into the CAN network, which must at least have one CAN device on the network.
3. Supply the 10~30 volts DC source into the I-7530 module through the power terminal.
4. The ON LED on the I-7530 module will be turned on. That means the I-7530 is working in the operation mode.
5. Run the I-7530 Utility software after there is a wire connection as shown in Section 2.5.
6. Select the "Test" tab and click the "Connect" icon on the tool bar in the I-7530 Utility to configure the PC COM port and Baud rate, which will be used to connect to the COM port of the I-7530.



7. Press the “OK” button. Then the I-7530 Utility will show the test tab in the window as can be seen in the below figure.



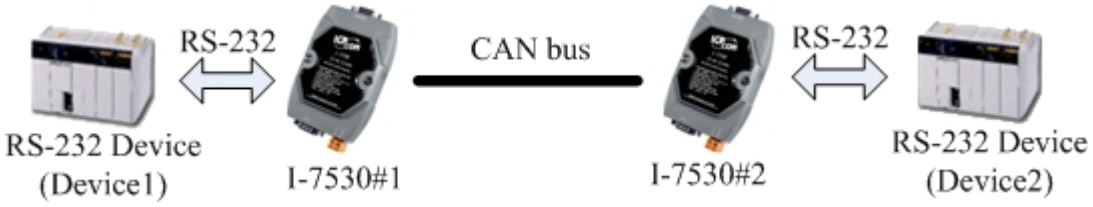
8. Users will then be able to transmit or receive CAN messages via the I-7530 module. There are two methods for communicating with their CAN. The first method (check “Use CAN Message”) requires users to understand their CAN specifications. And input the correct data in the CAN message format. The second method (uncheck “Use CAN Message”) allows the use of the command string found in table 5 to transmit messages. Both methods require the user to click the “Send” button to transmit the information into the CAN network. The CAN message will then automatically be displayed in the “Receive” list box. Besides, user can press the “Clear” button to remove the messages on the list box and then the new CAN messages will be shown on the list box again. Or user can press the “Save” button to save the CAN messages, in the “Receive” list box, into the “Receive.txt file”.

3.4 Pair Connection Mode Description

The pair connection function usually needs two I-7530s. When these two I-7530s are in pair connection mode, all RS-232 commands transmitted from one of these two I-7530s will be put in the data field of CAN message. This CAN message will be transferred to RS-232 commands by another I-7530. The following section will show each condition for different pair connection configuration.

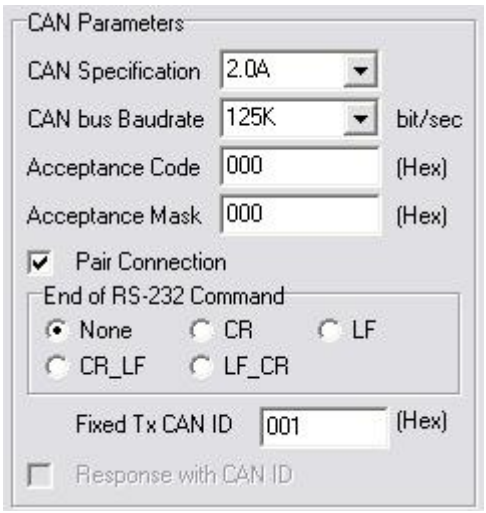
Application 1:

This application may be used in two general RS-232 devices which need to connect with each other, but the distance between is too large to communicate by using RS-232.

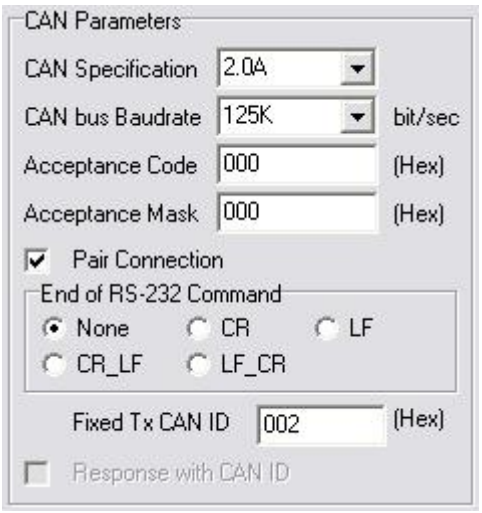


Configurations:

To apply this application, user need to configure the I-7530#1 and I-7530#2 as follows. The RS-232 configurations of I-7530#1 and I-7530#2 are decided by the Device1 and Device2 RS-232 parameters.



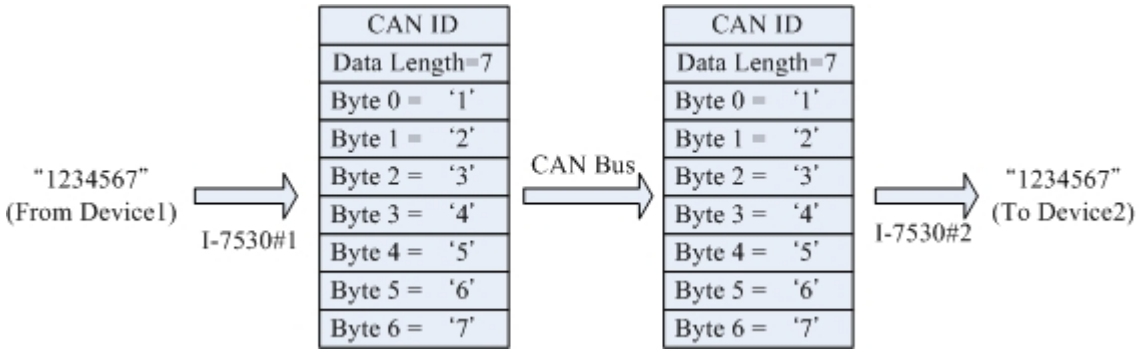
I-7530#1 Configuration



I-7530#2 Configuration

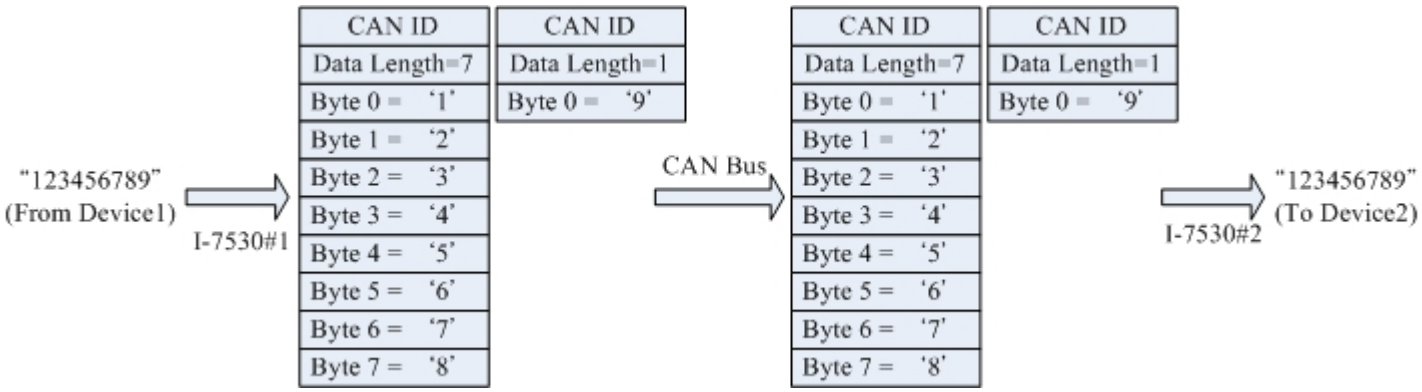
Communication Descriptions:

If there are 7 bytes data, “1234567”, transmitted from Device1, the Device2 will also receive “1234567” from the COM port of I-7530#2.



The CAN ID in above figure is determined by the CAN specification selected by users. If users select CAN 2.0A, the CAN ID is 11-bit ID. If CAN 2.0B is used, the CAN ID is 29-bit ID. Here, assume users set the Fixed Tx CAN ID field of I-7530#1 to be 0x001 (“0x” is for hexadecimal format) and CAN 2.0A is used, the CAN ID displayed in above figure is 0x001.

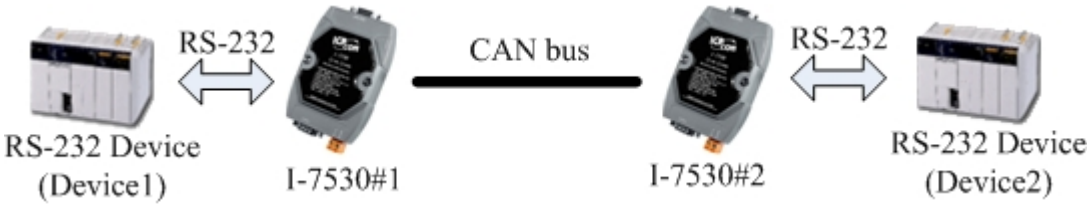
If there are 9 bytes data, “123456789”, transmitted from Device1, the Device2 will also receive “123456789” from the COM port of I-7530#2.



Note: If users use 115200bps for RS-232 port of I-7530, it is recommended that the configuration of I-7530 CAN baud is closed to the configuration of RS-232 baud, such as 125K bps. When you use pair connection function of I-7530, the baud under 125K bps is proper.

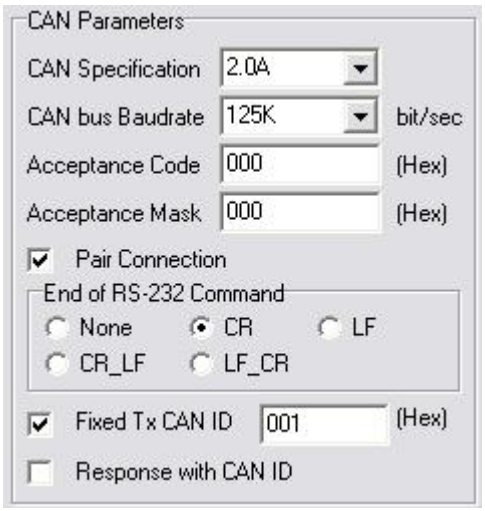
Application 2:

This application architecture is the same as the one of application1. The application architecture is show below. The difference will be discussed in the following paragraph.

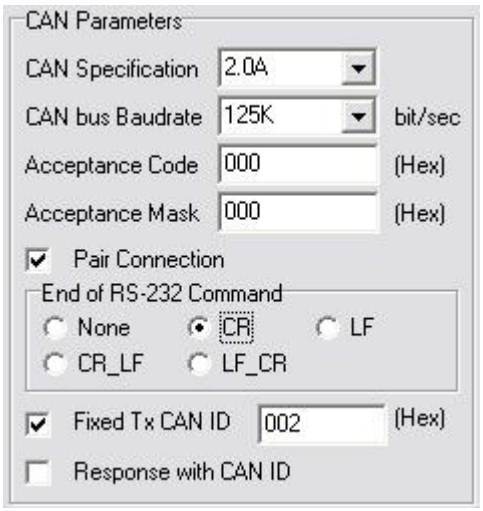


Configurations:

To apply this application, user need to configure the I-7530#1 and I-7530#2 as follows. The RS-232 configurations of I-7530#1 and I-7530#2 are decided by the Device1 and Device2 RS-232 parameters.



I-7530#1 Configuration



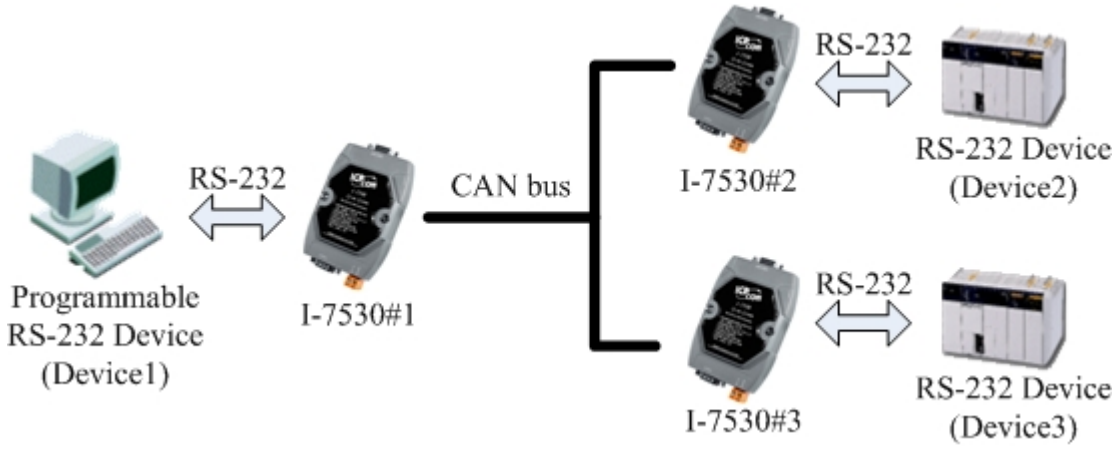
I-7530#2 Configuration

Communication Descriptions:

The communication of this condition is similar with the communication of condition 1. The difference is that the I-7530#2 of the application 1 will transfer the RS-232 commands to Device2 immediately if it receives any CAN message from the I-7530#1. The I-7530#2 of application 2 will not transfer the RS-232 commands to Device2 until it has checked the end character of RS-232 command (The end of RS-232 command is ‘CR’). For example, if Device1 send RS-232 commands “123456789”, the Device2 in application 1 will receive the data “12345678” immediately, and receive the data “9” with a little delay. But, Device2 in application 2 will receive the data “123456789” at the same time (Max. 72 bytes data at the same time).

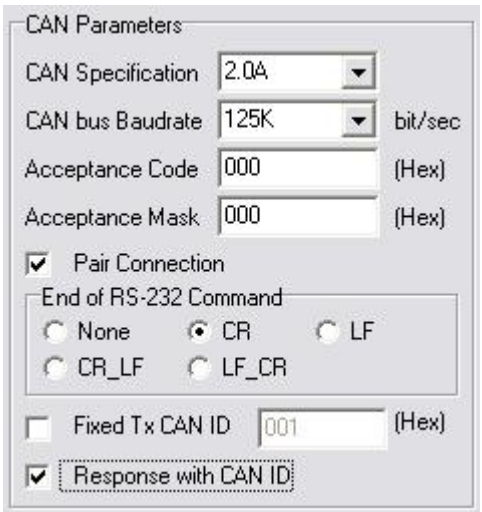
Application 3:

This application may be used to construct a RS-232 device network via CAN bus. The architecture is shown below.

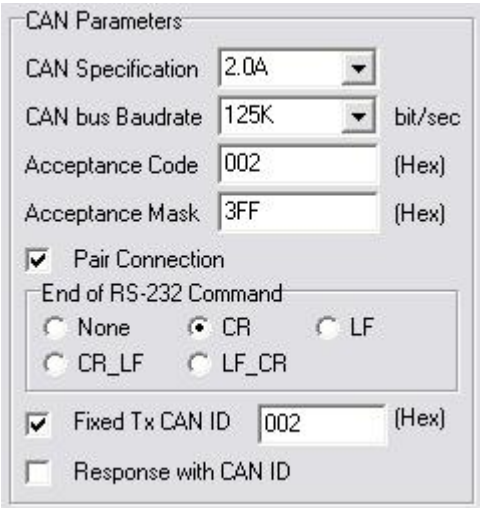


Configurations:

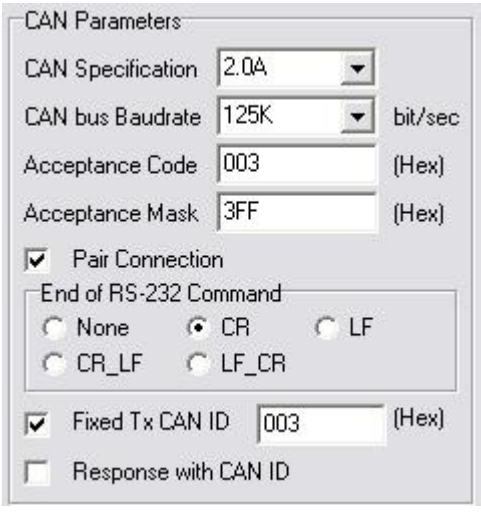
In order to apply this application, user need to configure the I-7530#1, I-7530#2, and I-7530#3 as follows. The RS-232 configurations of these three I-7530s are decided by the RS-232 device they connect with.



I-7530#1 Configuration



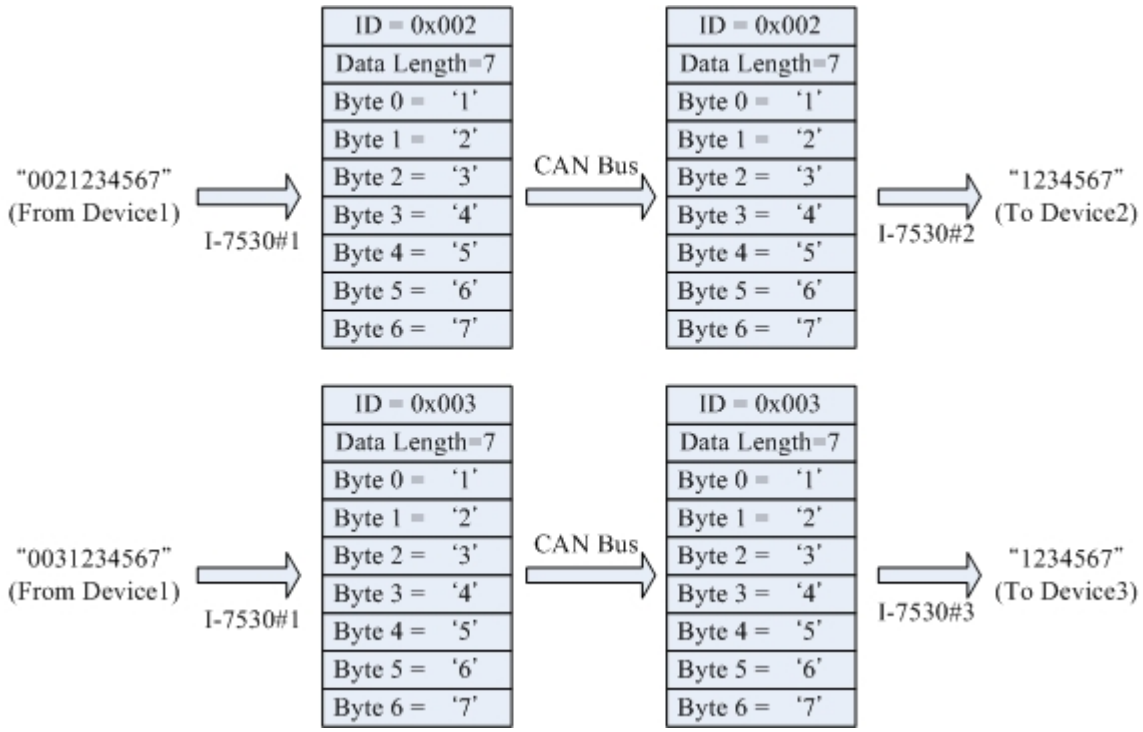
I-7530#2 Configuration



I-7530#3 Configuration

Communication Descriptions:

When Device1 want to transmit RS-232 commands “1234567” to Device2, the commands written to I-7530#1 by Device1 need to be “0021234567” because the Device1 is set to dynamic Tx CAN ID (Fixed Tx CAN ID is not checked). The first three bytes of “0021234567” is “002”, it means that the CAN ID is 0x002 when the I-7530#1 receives the RS-232 commands from Device1 and transfers it to CAN message. Afterwards, this CAN message is only accepted by Device2 because the configurations of acceptance code and acceptance mask of Device2. Similarly, if Device1 wants to send RS-232 commands “1234567” to Device3, it needs to send “0031234567” RS-232 commands to I-7530#1. When the Device2 or Device3 respond the RS-232 commands “456789”, the CAN message will have CAN ID “0x002” and “0x003” because of the configurations of Fixed Tx CAN ID of I-7530#2 and I-7530#3. Because the I-7530#1 is set to Response with CAN ID, the Device1 will receive the RS-232 commands “002456789” or “003456789”. Therefore, Device1 can decide the target device which RS-232 commands will be sent to. Also, Device1 knows where the RS-232 commands come from. The general concept of transmitting data from Device1 to Device2 is shown below.



Note: In pair connection mode, all RS-232 commands listed in section 4 are useless. When the RS-232 or CAN buffer of I-7530 is overflow. The ERR Led will be turned on for 300ms, and then I-7530 will be reset by watchdog automatically.

4. Command list

For easy application, we provide 9 command strings to allow users to send and receive commands and responses through the I-7530. It can cover most applications of different requests. The general formats of the commands for the I-7530 are in ASCII data format and given below::

Command Format: <Command>[CHK]<CR>

<Command> : The RS-232 commands of the I-7530 Converter.
[CHK] : 2-character checksum value. It is effective only if the checksum mechanism is set to enable by using I-7530 Utility. For checksum algorithm, please refer to page 27
<CR> : All RS-232 commands of the I-7530 Converter must end with the character “<CR>” (The ASCII value is 13).

The 9 command formats are given in the following table. More detailed information related to of the each command will be described in the following sub sections.

Table 5: Command list table

Command	Description
tIIILDD…[CHK]<CR>	Send or receive a standard data frame.
TIIIL[CHK]<CR>	Send or receive a standard remote frame.
eIIIIIIIILDD…[CHK]<CR>	Send or receive an extended data frame.
EIIIIIIIIL[CHK]<CR>	Send or receive an extended remote frame.
S[CHK]<CR>	Read the status value of I-7530
C[CHK]<CR>	Clear CAN/RS232 FIFO error flag of overflow.
P0BBDSPAE[CHK]<CR>	Change the RS-232 configuration
P1B [CHK]<CR>	Change the CAN configuration
RA[CHK]<CR>	Reboot the I-7530 module.

Checksum algorithm:

The checksum [CHK] is 2-characters of the sum of the command message, from first character to the character before <CR>. When calculate checksum, the value of [CHK] is zero.

For example:
Command: Reboot the I-7530 module, “RA[CHK]<CR>”.

1. Sum of the string = ‘R’ + ‘A’ = 52h + 41h = 93h.
2. Therefore the checksum is 93h and so [CHK]=”93”.
3. The command string with checksum =”RA93<CR>”.

4.1 tIIILDD…[CHK]<CR>

Description: Send or receive a standard CAN data frame.

- **Syntax: tIIILDD…[CHK]<CR>**

t	Represent a standard (2.0A) data frame.
III	11 bits Identifier (000~7FF)
L	Data length (0~8)
DD…	Input data frame value according to the data length (00~FF)

- **Response:** Valid command: No response
  Invalid command: ?<Error Code><CR>
- **Note:** It is necessary to enable the "Error Response" function in the I-7530 Utility, in order to receive Syntax and/or communication error information at the host PC.

- **Example:**
  Command: t03F6112233445566<CR>
  Send a CAN message with a standard data frame. ID=03F, DLC=6, data1=11, data2=22, data3=33, data4=44, data5=55 and data6=66.

4.2 TIIIL[CHK]<CR>

Description: Send or receive a standard CAN remote frame.

- **Syntax: TIIIL[CHK]<CR>**

T	Represents a standard (2.0A) remote frame.
III	11 bits Identifier (000~7FF)
L	Data length (0~8)

- **Response:** Valid command: No response
  Invalid command: ?<Error Code><CR>
- **Note:** It is necessary to enable the "Error Response" function in the I-7530 Utility, in order to receive Syntax and/or communication error information at the host PC.

- **Example:**

  Command: T2E88<CR>

  Send a CAN message with a standard remote frame. ID=2E8, DLC=8.

4.3 eIIIIIIIILDD…[CHK]<CR>

Description: Send or receive an extended CAN data frame.

- **Syntax: eIIIIIIIILDD…[CHK]<CR>**

e	Stands for the extended (2.0B) data frame.
IIIIIIII	29 bits Identifier (00000000~1FFFFFFF)
L	Data length (0~8)
DD…	Input data frame value according to the data length (00~FF)

- **Response:** Valid command: No response
  Invalid command: ?<Error Code><CR>
- **Note:** It is necessary to enable the “Error Response” function in the I-7530 Utility, in order to receive Syntax and/or communication error information at the host PC.

- **Example:**

  Command: e123456785112233445 5<CR>

  Send a CAN message with an extended data frame. ID=12345678, DLC=5, data1=11, data2=22, data3=33, data4=44 and data5=55.

4.4 EIIIIIIIIL[CHK]<CR>

Description: Send or receive an extended CAN remote frame.

- **Syntax: EIIIIIIIIL[CHK]<CR>**

E	Stands for the extended (2.0B) CAN remote frame.
IIIIIIII	29 bits Identifier (00000000~1FFFFFFF)
L	Data length (0~8)

- **Response:** Valid command: No response
  Invalid command: ?<Error Code><CR>
- **Note:** It is necessary to enable the “Error Response” function in the I-7530 Utility, in order to receive Syntax and/or communication error information at the host PC.

- **Example:**
  Command: E010156786<CR>
  Send a CAN message with an extended remote frame. ID=01015678, DLC=6.

4.5 S[CHK]<CR>

Description: Read the I-7530 CAN Baud Rate and error flag message.

- **Syntax: S[CHK]<CR>**

S	Command character.

- **Response:** Valid Command: !CFFTTRRO[CHK]<CR>
  Invalid command: ?<Error Code>[CHK]<CR>

!	Delimiter for valid command
C	current baud rate setting of CAN
FF	CAN register
TT	CAN transmit error counter
RR	CAN receive error counter
O	CAN/RS-232 FIFO Overflow flag

- Note**:** It is necessary to enable the “Error Response” function in the I-7530 Utility, in order to receive Syntax and/or communication error information at the host PC. Furthermore, all response results are shown in the ASCII format. Users need to make an ASCII to hex format transformation in order to understand what the meaning is based on the following table.

Table 6: CAN baud rate list

AsciiToHex(C)	Description
0	10K baud rate of CAN
1	20K baud rate of CAN
2	50K baud rate of CAN
3	100K baud rate of CAN
4	125K baud rate of CAN
5	250K baud rate of CAN
6	500K baud rate of CAN
7	800K baud rate of CAN
8	1000K baud rate of CAN

Table 7: CAN register list

AsciiToHex(FF)	Description
Bit 7	Bus Off Mode
Bit 6	Error Passive Mode
Bit 5	Reserved
Bit 4	Overrun Buffer
Bit 3	Stuff Error General
Bit 2	CRC Error General
Bit 1	Form Error General
Bit 0	Acknowledgment Error General

Table 8: CAN/RS-232 FIFO overflow flag list

AsciiToHex(O)	Description
Bit 3	Reserved
Bit 2	Reserved
Bit 1	RS-232 FIFO Overflow
Bit 0	CAN FIFO Overflow

- **Example:**

  Command: S<CR>

  Receive: !50000000<CR>

  Obtain some current information on the I-7530 module. The response will show the following results: CAN baud rate=250K, CAN register= normal, CAN transmit error counter=0, CAN receive error counter=0 and CAN/RS232 FIFO= normal.

4.6 C[CHK]<CR>

Description: Clear the CAN/RS-232 FIFO overflows error flag on the module.

- **Syntax: C[CHK]<CR>**

  C Command character.

- **Response:** Valid Command: No response. This will turn the ERR LED off after the ERR LED was turned on as a result of the CAN/RS232 FIFO overflow error flag.
  Invalid command: ?<Error Code>[CHK]<CR>
- **Note:** It is necessary to enable the "Error Response" function in the I-7530 Utility, in order to receive Syntax and/or communication error information at the host PC.

- **Example:**
  Command: C<CR>

4.7 P0BBDSPAE[CHK]<CR>

Description: Change the RS-232 configuration on the I-7530 module and then reboot the I-7530 module.

- **Syntax: P0BBDSPCE[CHK]<CR>**

P0	Command character
BB	RS-232 Baud rate
D	Data bit 0 = 5 bits Data formation 1 = 6 bits Data formation 2 = 7 bits Data formation 3 = 8 bits Data formation
S	Stop bit (0=1 stop bit, 1=2 stop bits)
P	Parity (0=None, 1=Odd, 2=Even)
C	Checksum (0=No, 1=Yes)
E	Error Response (0=No, 1=Yes);

Table 9: RS-232 baud rate list

BB	Description
00	110 bps baud rate of RS-232
01	150 bps baud rate of RS-232
02	300 bps baud rate of RS-232
03	600 bps baud rate of RS-232
04	1200 bps baud rate of RS-232
05	2400 bps baud rate of RS-232
06	4800 bps baud rate of RS-232
07	9600 bps baud rate of RS-232
08	19200 bps baud rate of RS-232
09	38400 bps baud rate of RS-232
0A	57600 bps baud rate of RS-232
0B	115200 bps baud rate of RS-232

- **Response:** A valid command will write the RS-232 configuration parameters into the EEPROM and then reboot the I-7530 module.
  Invalid command: ?<Error Code><CR>
- **Note:** It is necessary to enable the “Error Response” function in the I-7530 Utility, in order to receive Syntax and/or communication error information at the host PC.

- **Example:**
  Command: P00B30000<CR>
  Set the RS232 baud rate=115.2K, data bit=8, stop bit=1, none parity, no checksum and No error responses into the I-7530 module and then reboot the I-7530 module.

4.8 P1B [CHK]<CR>

Description: Change the CAN configuration of I-7530 module and then reboot the I-7530 module.

- **Syntax: P1B[CHK]<CR>**

  P1 Command character
  B CAN Baud rate

Table 10: CAN baud rate list

B	Description
0	10K baud rate of CAN
1	20K baud rate of CAN
2	50K baud rate of CAN
3	100K baud rate of CAN
4	125K baud rate of CAN
5	250K baud rate of CAN
6	500K baud rate of CAN
7	800K baud rate of CAN
8	1000K baud rate of CAN

- **Response:** A valid command will write the CAN configuration baud rate into the EEPROM and then reboot the I-7530 module.
  Invalid command: ?<Error Code><CR>
- **Note:** It is necessary to enable the “Error Response” function in the I-7530 Utility, in order to receive Syntax and/or communication error information at the host PC.

- **Example:**
  Command: P14<CR>
  Set the CAN baud rate=125K into the I-7530 module and then reboot the I-7530 module.

4.9 RA[CHK]<CR>

Description: Reboot the I-7530 module. If the module is displaying on but the CAN bus is off, users can use this command to reboot the module in order to allow it to work in order again.

- **Syntax: RA[CHK]<CR>**

  RA Command character

- **Response:** Valid command will reboot the I-7530 module.
  Invalid command: ?<Error Code><CR>
- **Note:** It is necessary to enable the "Error Response" function in the I-7530 Utility, in order to receive Syntax and/or communication error information at the host PC.
- **Example:**
  Command: RA<CR>
  The I-7530 module will reboot after it had received this command.

4.10 General Error code for all command

Table 11: Error code table

AsciiToHex(Error code)	Description
1	The head character of the command string is invalid.
2	The length of the command string is invalid.
3	The checksum of the command string is invalid.
4	Reserved
5	The RS-232 receiver is timeout.

5. Troubleshooting

If the Error response function on the I-7530 module is set to be “Yes”(that means enable) via the I-7530 Utility during the configuration period, the I-7530 will send the error code to the RS-232 device of the host PC through the RS-232 media when the I-7530 produces an error message automatically during the operation mode. The meanings of these error codes are given below:

Table 12: Error code table

Error code	Description	Possible causes & solutions
1	Invalid header	The RS-232 command string header is not “t”,”T”,”e”,”E”,”S”,”C”,”P0”, ”P1” nor “RA”.
2	Invalid length	The data byte of the CAN Message does not match the data length of the CAN Message. For example: Error: t001512345<CR> Right: t00150102030405<CR>
3	Invalid checksum	The checksum from the RS-232 command string does not matched with the checksum calculated by the I-7530. For example: Error: t0012112209<CR> Right: t00121122FD<CR>
5	Timeout	The ASCII command strings are sent incomplete. For example: Error: T0018 Right: T0018<CR>

If the I-7530 CAN baud rate is not the same as the CAN baud rate on the CAN network, the ON LED on the I-7530 will be flash with a constant frequency because the I-7530 cannot send any CAN messages to the CAN network. Therefore, users will need to read the I-7530 status by using the command “S[CHK]<CR>”(in the section 4.5) to help users understand what is going in the module. In general, the following errors could occur: CAN media connection problem, terminal resistor problem, different baud rate configuration with CAN network and the like.